POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Diagnostic Text as Supervision: Language-Conditioned Learning for Semantic Segmentation

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Luca Olivieri

Advisor: Prof. Matteo Matteucci

Co-advisors: Nico Catalano, Sara Pidò, Chiara Plizzari

Academic year: 2024-2025

1. Introduction

Deep Learning (DL) has achieved remarkable success in Semantic Segmentation (SS), enabling models to learn complex visual patterns and achieve state-of-the-art performance. However, conventional training relies on numerical losses that may fail to promote robust, semantically grounded representations, encouraging reliance on spurious correlations and limiting generalization.

Building on this perspective, Human-In-The-Loop (HITL) approaches can integrate human knowledge into deep learning to guide models toward robust, generalizable representations instead of superficial patterns, providing semantically meaningful guidance. However, current approaches either underutilize the full expressive potential of language or focus mainly on inference, leaving training-time HITL underexplored.

To address these limitations, this work proposes a novel training paradigm for SS that incorporates language-based **diagnostic feedback**. Instead of relying solely on numerical losses, multimodal language models generate interpretable descriptions of prediction errors (e.g., inaccurate boundaries or merged regions). These textual diagnostics are integrated via a contrastive loss, which guides the segmentation model to align visual features with the descriptions, grounding them in meaningful spatial properties.

A comprehensive experimental pipeline is developed to evaluate this **language-guided training strategy** through systematic experiments that analyze the impact of diagnostic text on learning dynamics and segmentation performance. Key contributions include a class-split diagnostic text generation setup, a grouped contrastive loss implementation that balances the contribution of each class, and a subsampling mechanism to improve the efficiency of the language-guided training.

This study highlights both the potential and limitations of linguistic guidance in spatial vision, showing that jointly optimizing pixel predictions and text alignment is challenging and can degrade performance and stability.

2. Background

Language-guided supervision, while still underexplored, has produced a few notable results. [3] proposes guiding a frozen segmentation network at inference time using natural language hints encoded by a GRU. The text embeddings are processed into scaling and bias parameters

that modulate intermediate activations through lightweight guiding blocks. The backbone remains frozen, while the GRU and guiding layers are trained from scratch separately for each image. Although flexible, the method applies language supervision only at inference time, neglecting the training phase. Also, since guidance is optimized per image, it lacks transferable language–vision representations and risks overfitting, limiting generalization and making extension to training-time supervision non-trivial. [2] proposes supervising vision encoders with high-level language specifications to guide spatial attention during training. They add an auxiliary loss that aligns the classifier's attention maps with those derived from CLIP. CLIP attention maps are computed by applying Grad-CAM to the activations generated on a set of templated texts (e.g., "an image of class"), while classifier attention maps are obtained via input gradients. The loss encourages the model to focus on regions highlighted by the vision–language model. While effective, the method relies on simple synthetic prompts and coarse-grained concepts. As a result, it does not fully leverage the expressive richness of natural language, leaving deeper forms of language supervision unexplored.

3. Methods

3.1. Diagnostic Text Generation

The proposed approach relies on textual descriptions encoding diagnostic information, requiring an automated text generation mechanism. To this end, a **Multi-modal Large Language Model (MLLM)** is employed to produce diagnostic reports that describe how the predicted segmentation mask deviates from the ground truth, identifying error regions and assessing overall quality. For each sample, the model receives a class-specific prompt containing task instructions, the predicted mask, the ground-truth mask, and a set of human-annotated demonstrations to guide generation.

Class-split Prompting Strategy. The prompting strategy adopts a class-specific design in which multi-class segmentation masks are decomposed into binary masks, each associated with a single target class, while all others are treated as negative. This removes the need to interpret color maps or infer class identities, reducing hallucinations and improving accuracy. Ground-truth and predicted masks are provided as RGB images, either as separate images or concatenated, and overlaid on the original image to preserve context while maintaining clear class separation. In non-class-split settings, a color map must be supplied so the MLLM can interpret class identities. The color map can be provided in the form of an image, textual color names, or class-specific patches.

LLM-as-a-Judge Evaluation. The MLLM's performance is assessed through an LLM-as-a-Judge approach: a separate Large Language Model (LLM) evaluates whether each predicted diagnostic text is correct relative to the ground truth, producing a binary decision. In class-split scenarios, accuracy is computed per class and averaged across samples. For validation, 80 ground-truth diagnostic texts from VOC2012 were manually annotated, and predictions were generated using LR-ASPP-MobileNetV3-Large.

3.2. Diagnostic Text Encoding

The diagnostic texts generated by the MLLM are fed to a **Vision-Language Encoder (VLE)** to generate embeddings used as supervisory signals in the training pipeline. To improve the VLE's ability to capture diagnostic text information, a checkpoint of the FLAIR VLE [1] is fine-tuned on Synthetic Diagnostic Dataset (SDD).

SDD is a class-split dataset of diagnostic mask–text pairs designed to enhance the contrastive capabilities of the VLE. It is synthetically generated from all samples in VOC2012 and COCO2017 (standardized to VOC2012 classes). Binary diagnostic masks mark pixels where predictions disagree with their ground truth and are overlaid on the background. These masks are class-split and paired with diagnostic texts generated by Gemma 3 12B QAT using the prompting procedure described in Subsection 3.1. The dataset contains 184,017 training and 10,579 validation samples, preserving the class distribution of the source datasets.

The FLAIR text encoder was fine-tuned on SDD using an adapter-based approach, where a trainable text adapter projects the encoder outputs into a new space while keeping the original encoder frozen. Training is guided by the original

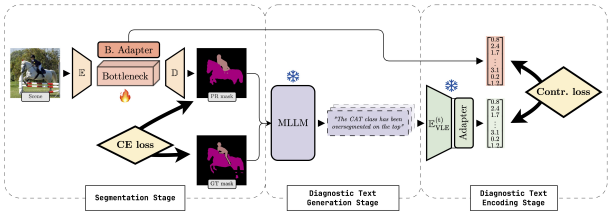

Figure 1: An overview of the language-guided pipeline.

FLAIR loss [1].

3.3. Language-guided Pipeline

The training pipeline, illustrated in Figure 1 extends standard segmentation training frameworks by incorporating diagnostic text generation and encoding modules, which provide language-based training supervision.
A frozen MLLM generates class-specific diagnostic texts from the predicted and ground-truth masks. Such texts are then encoded by a frozen VLE text encoder. The bottleneck features of the segmentation encoder are projected through a bottleneck adapter into an embedding space shared with the text representations. A contrastive loss then aligns these visual features with the corresponding text embeddings, encouraging the encoder to learn text-aligned representations. In parallel, a Cross-Entropy (CE) loss supervises the predicted masks. Training optimizes a multi-task objective combining the CE and contrastive losses, either as an **auxiliary** term (Eq. 1a) or in a **convex combination** (Eq. 1b):

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{CL}}, & \text{(aux.)} \quad \text{(1a)} \\ (1-\lambda)\mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{CL}}, & \text{(convex)} \quad \text{(1b)} \end{cases}$$

Grouped Contrastive Loss. The variable number of class-specific diagnostic texts per sample introduces imbalance when applying standard contrastive losses, as samples with more texts would dominate the optimization. To address this, a **Grouped SigLIP loss**, which aggregates positive and negative visual–text alignments at the sample level, is used. Each item's loss is computed via standard sigmoid-based terms, summed with its negatives, then normalized by the group size and averaged across all samples, ensuring equal contribution regardless of the text count. Bottleneck vectors can be shared across class-split texts for simplicity or tailored per class for finer alignment.

Synthetic Contrastive Negatives. Beyond standard in-batch negatives, synthetic per-image textual negatives can be used, giving each positive pair a fixed, controllable set of negatives and potentially harder examples. For each bottleneck feature, negatives are generated from its positive diagnostic text through a three-stage process: (A) **random word substitution** using predefined word pools to create semantically coherent variants, and fallback steps (B) **random word shuffling** and (C) **random word removal** if more negatives are needed.

Data Subsampling Strategy. To reduce the computational cost of MLLMs during training, an input subsampling strategy is applied, selecting only the most informative samples for language-guided supervision. Only classes present in the ground truth are considered, and among these, only the top-p masks exhibiting the largest temporal changes are retained via a cache that tracks previous predictions. This prioritizes samples that most influence learning while discarding stable predictions.

Bottleneck Adapters. Bottleneck adapters mediate the interaction between the segmentation model's feature volume and the diagnostic text features by projecting both into a shared embedding space. Three main types are used: (A) **linear-based** modules that apply Global Average Pooling (GAP) followed by linear layers, discarding spatial information early; (B) **convolution-based** modules that process spatial structure via point-wise convolutions before the GAP pooling; and (C) **attention-based** modules that use learnable attention pooling, optionally conditioned on the class-specific text vectors for explicit bottleneck class-conditioning. All adapters introduce trainable parameters optimized through the contrastive loss.

4. Experiments

4.1. Preliminary Experiments on the MLLM

The evaluated MLLMs underwent a testing phase to measure their ability to generate diagnostic text.

Evaluation Prompt Selection. To align the LLM-as-a-Judge with human evaluations, multiple evaluation prompt variants were tested using 20 ground-truth assessments in both non–class-split and class-split settings, employing Gemini 2.5 Flash as the judge model. In the non–class-split scenario, a base prompt was augmented with combinations of instructions targeting different evaluation criteria. In the class-split scenario, the prompt was extended with explicit instructions to restrict evaluation to a specified positive class. The alignment was evaluated by computing the accuracy between ground-truth and predicted evaluations.

Inference Prompt and MLLM Selection. Using the LLM-as-a-Judge setup selected in the prior phase, a series of experiments assessed the impact of inference prompt design and model choice on diagnostic text generation. A lightweight prompt template with limited explicit instructions and optional few-shot examples was adopted to balance expressiveness and output stability. First, class-split and non-class-split settings were compared. Further experiments evaluated mask formatting, the number of few-shot examples (0–3), and different MLLMs. The performance was evaluated by computing the percentage of correct predictions.

4.2. Preliminary Experiments on FLAIR

The FLAIR fine-tuning was conducted on SDD by training a lightweight adapter, a single linear layer, on top of the frozen text encoder outputs, keeping all original parameters fixed. Training was performed using a protocol aligned with the original FLAIR setup [1]. Model capability was assessed through training metrics and qualitative inspection of attention maps generated as in FLAIR [1].

4.3. Language-Guided Pipeline Experiments

The behavior of the language-guided training pipeline is assessed in different scenarios. All models were trained and validated on the original VOC2012 splits and undertook a pre-language training phase, with the cache prefilled at the beginning of training. For efficiency, the validation contrastive loss was not computed.
Main experiments analyze the contribution of (1) λ values, (2) synthetic negatives, (3) bottleneck adapters, (4) segmentation models, and (5) the data subsampling strategy. Additional experiments ablate the contribution of (6) FLAIR SDD fine-tuning, (7) the subsampling strategy (also investigating the validation contrastive loss), and (8) the segmentation encoder on the language-guiding process. Experiments (1-3), (5), (7) were conducted with a multi-task loss setup, with the others using an auxiliary setup. The evaluation metrics include the mean Intersection over Union (mIoU), the CE and contrastive loss values, and the gradient norm magnitude as a measure of training stability.

5. Results

5.1. Preliminary Results

In this subsection, the results of the preliminary MLLM and FLAIR experiments are presented.

LLM-as-a-Judge Evaluation Prompt. In the non-class-split scenario, a set of instructions regarding precision, error categorization, and spatial awareness gave the highest accuracy (85%) and was selected. In the class-split scenario, the same prompt was successfully adapted to constrain the evaluation to the positive class.

MLLMs and Inference Prompt. The results indicate class-split prompts outperform non-class-split (74.0% vs 66.5% accuracy, obtained with patches color map). Evaluation across models, mask formats, and shots shows that Gemma 3 27B achieves the top performance, Gemma 3 12B QAT strikes the best trade-off between model size and performance, Mistral S 3.1 24B showcases inconsistent performance, while Qwen2.5-VL 7B underperforms. Separate and concatenated masks yield similar results. Few-shot prompting improves performance, but improvements do not systematically scale with more shots. Based on these findings, Gemma 3 12B QAT was selected for generating diagnostic texts, with separate masks and one-shot prompting.

FLAIR Fine-tuning. Training trends show contrastive loss dropping sharply in early epochs before rapidly plateauing and converging to a final value of 5.1833. Validation loss follows a similar pattern but settles slightly higher at 5.3576, indicating mild overfitting. The training and

validation losses confirm a marked improvement in encoding capabilities.

5.2. Language-Guided Pipeline Results

In this subsection, the results of the language-guided pipeline experiments are presented.

(1, 2) Lambda and Synthetic Negatives. As shown in Figures 2 and 3, increasing the contrastive weight λ progressively degrades segmentation performance while improving contrastive loss optimization, but introduces larger gradient norms and training instability, without yielding generalization gains over the baseline.
Replacing in-batch negatives with synthetic negatives further worsens both segmentation and contrastive metrics, especially for higher λ, and amplifies optimization instability.

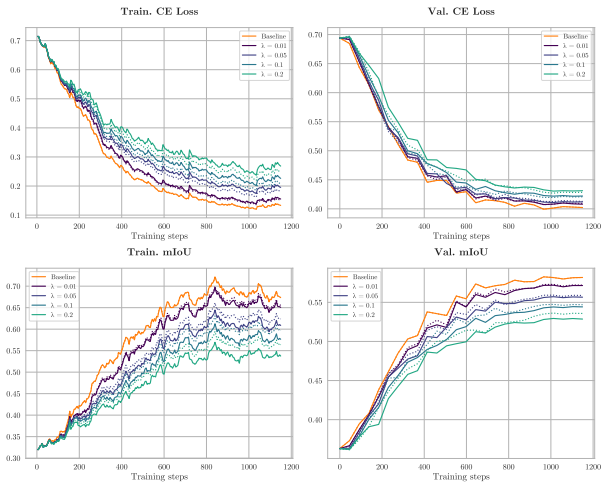

Figure 2: Train./val. segmentation metrics for lambda and synthetic negatives experiments. Dotted lines: in-batch negatives.

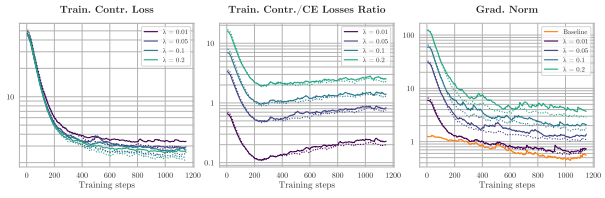

Figure 3: Train./val. contrastive and gradient metrics for lambda and synthetic negatives experiments. Dotted lines: in-batch negatives.

(3) Bottleneck Adapters. As visible in Figures 4 and 5, bottleneck adapter design has a significant impact: linear adapters perform worst, convolutional adapters improve both tasks, and attention-based adapters (especially `conv+AttnPoolByText`) achieve the most effective segmentation and contrastive optimization, albeit with higher gradient norms. Segmentation performance benefits from stronger adapters that more effectively optimize the contrastive loss.

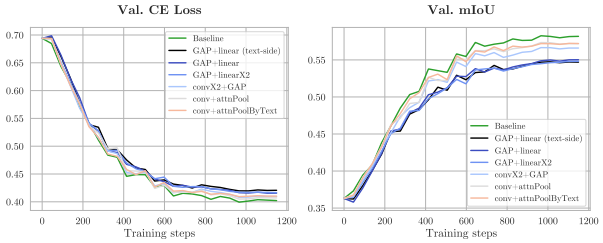

Figure 4: Val. segmentation metrics for the bottleneck adapters experiment.

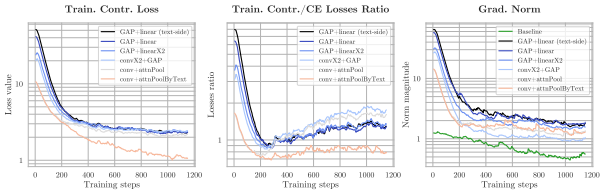

Figure 5: Train. contrastive and gradient metrics for the bottleneck adapters experiment.

(4) Segmentation Models. The results, shown in Table 1, indicate that, when varying the underlying segmentation architectures, the language-guided mechanism does not lead to systematic performance improvement across architectures, with only marginal validation gains in the FPN configurations.

(5) Data Subsampling. More aggressive data subsampling (lower p) consistently reduces both segmentation and contrastive performance, and leads to markedly higher gradient instability. This suggests reduced data diversity increases optimization difficulty and degrades performance.

(6) FLAIR Fine-tuning Ablation. Ablating the SDD fine-tuning of FLAIR shows slight improvements in segmentation metrics and more stable training, but contrastive optimization is negatively affected. This indicates an inverse relationship between segmentation and contrastive objectives.

	CE Loss		mIoU		Contr. Loss
	Train.	Val.	Train.	Val.	Train.
LR-ASPP-MobileNetV3	0.1360 (+0.046)	0.4239 (+0.024)	73.25 (-10.58)	54.84 (-3.42)	0.478
FPN-ResNet18	0.0509 (+0.022)	0.5122 (-0.024)	85.84 (-5.80)	52.88 (+0.72)	0.346
DeepLabV3-ResNet18	0.0627 (-0.065)	0.3877 (-0.063)	92.86 (+0.19)	59.77 (-1.00)	0.390
FPN-ResNet34	0.0330 (+0.013)	0.4674 (+0.002)	89.74 (-5.07)	58.74 (+0.07)	0.345
DeepLabV3-ResNet34	0.1402 (+0.030)	0.4326 (-0.016)	90.99 (-2.59)	62.24 (-1.86)	0.779

Table 1: Optimal CE loss, mIoU, and contrastive losses for segmentation models, relative to baseline. Green → better, red → worse.

(7) Subsampling Ablation and Validation Contr. Loss. Disabling subsampling ($p = 1.0$) slightly improves segmentation metrics relative to the default $p = 0.4$, although still remaining below the baseline. Contrastive training shows low overfitting, while gradient norms are lower than with partial subsampling, yet still higher than baseline. These results indicate that moderate subsampling offers a good trade-off between efficiency and performance, while the alignment task remains challenging even with the strongest bottleneck adapter.

(8) Seg. Encoder Detachment Ablation. The outcomes, illustrated in Figure 6, highlight that detaching the segmentation encoder from the contrastive loss leaves contrastive optimization largely unchanged, with only a slight loss increase and similar trends, while improving stability. This indicates that the bottleneck adapter alone can effectively handle most of the contrastive optimization.

Figure 6: Train. contrastive loss and gradient norms for the seg. encoder detachment ablation study.

6. Conclusions

The results indicate that the proposed method does not systematically outperform baseline segmentation models. Joint optimization leads to instability, evidenced by decreased mIoU and higher gradient norms, suggesting competition between the contrastive and CE objectives for the encoder capacity. Stronger bottleneck adapters mitigate this issue by handling most of the linguistic alignment, thereby reducing interference with segmentation, as supported by the bottleneck adapter studies. Further experiments reveal that greater encoder participation in contrastive supervision intensifies its detrimental effect on segmentation performance, highlighting that segmentation encoders are ill-suited to jointly optimize pixel-level classification and diagnostic text alignment.

Moreover, while moderate subsampling enhances the efficiency while preserving training dynamics, aggressive subsampling and synthetic negatives amplify instability and further complicate the contrastive objective optimization.

These findings reveal a gap between segmentation and linguistic representations, emphasizing the need for innovative, synergistic multi-modal approaches that encourage cooperation between pixel-level and textual objectives.

References

[1] Xiao at al. Flair: Vlm with fine-grained language-informed image representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

[2] Petryk et al. On guiding visual attention with language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[3] Rupprecht et al. Guide me: Interacting with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.